DESIGN AND SIMULATION OF INFINITE IMPULSE RESPONSE (IIR) FILTERS USING THE IMPULSE INVARIANT METHOD WITH MATLAB GRAPHICAL USER INTERFACE (GUI)

Nauval Rusydi[1], Redi Ratiandi Yacoub[2], Eka Kusumawardhani[3]
nauval.rusydi@gmail.com[1], rediyacoub@ee.untan.ac.id[2], ekawardhani@ee.untan.ac.id[3]
Universitas Tanjungpura

ABSTRACT
Matlab is a software that is capable of performing calculations and simulations as well as developing applications based on Graphical User Interface (GUI). GUI-based application development can be done for digital signal processing, one of which is IIR digital signal filtering. The purpose of this research and development is to design a GUI to simulate an IIR digital filter. By entering the signal to be filtered and providing input parameter values in the form of filter type, filter order, cutoff frequency, passband ripple, and stopband ripple, the GUI will process the input values to filter the input signal given into an output in the form of a frequency spectrum. This GUI will be made stand-alone so that users do not need to open the Matlab application to use it. Testing for IIR digital signal filtering from the GUI is carried out by a digital filter verification test by comparing using various different cutoff frequencies, while maintaining other parameters where the results of the output signal have the same characteristics according to the filter. Then, to test the usefulness of each button, a blackbox test is carried out where the results obtained are that all buttons can work according to their respective functions without any problems or errors so that the resulting GUI can be used for IIR digital signal filtering. Then, for testing the GUI display, a survey was conducted on 30 respondents where the results had a value of 4.26 with the statement very satisfied.
Keywords: *IIR Filter, GUI, Filter Response, Stand-alone GUI, Simulation.*

INTRODUCTION

The development of technology today is very rapid, for example in the field of communication. In communication, signals play an important role, namely as a carrier of information. In the process of sending information there is always interference or noise, the nature of noise cannot be eliminated but can be minimized [1]. In order for the signal to be in accordance with needs and desires, a filter is needed to sort the signal from noise. A filter is a design to pass or filter input signals so that the incoming signal matches the desired frequency. The filter system is very important in the communication system, namely the sender and receiver sides [2]. For this reason, this study will create a GUI that is used to filter signals.

Digital filters have 2 types, one of which is IIR (Infinite Impulse Response). IIR digital filters have infinite impulse response, so they can be matched with analog filters that also have infinite duration impulse response [3]. IIR Filter is a type of digital filter, which can be used to perform all types of filtering, namely high-pass, low-pass, band-pass and band-reject. The filter functions as a selector of the desired frequency [4]. In designing a filter, a design simulation can be used through software that aims to facilitate the determination of the circuit design that will be used to assemble a filter and can explain the characteristics and response of a filter. One of the software that can perform this design simulation is Matlab.

Currently, the use of Matlab as a learning tool has been widely used in various places, especially at the university level. Such as at the University of Sriwijaya, South Sumatra, which conducted research on the effects of applying the windowing method to the FIR LPF to determine the performance of various types of windowing. In this study, researchers will create a GUI that will be used to filter audio signals by providing the required input parameter values and producing Butterworth, Chebyshev I, and Chebyshev II filter type responses, and Elliptic with lowpass, highpass, bandstop, and bandpass filter types.

LITERATURE REVIEW

There are similar studies that have existed previously so that they can be used as input in this study. Each of these studies provides insights and methodologies that can be applied directly to the development of a Matlab GUI for IIR signal filtering.

Research conducted by Murdika, et al [5]. This research contains this study designs and simulates IIR filters using Graphical User Interface (GUI)". This study uses the Pole-zero placement, Impulse invariant, Matched z-transform, and Bilinear z-transform methods in realizing IIR filters and for example to design low-pass, high-pass and band-pass filters. This study contains the design and simulation of IIR filters that have inputs that aim to filter audio signal files that are entered into the GUI. After the audio signal is entered, the GUI will filter so that the output of the audio signal becomes better.

Research conducted by Aditya Girza Utama, et al [6]. This research contains creating a Matlab graphical user interface (GUI) for demonstrating the windowing method on a low pass filter (LPF) finite impulse response (FIR)". This research has designed a low pass filter (LPF) FIR using the Rectangular, Hamming and Hanning windowing methods. This study compares the performance of various types of windowing and provides an understanding of the effects of windowing methods on the FIR LPF.

Research conducted by Sri Rahayu Islamiyah, et al [1]. This research contains Implementation of Bilinear Transformation Method on Infinite Impulse Response (IIR) Digital Filter Using Raspberry Pi. This research uses bilinear transformation method with butterworth filter response, chebyshev type 1 and chebyshev type 2 in MATLAB. This research provides guidance on how to implement IIR digital filter on raspberry pi, and shows the results of IIR digital filter.

Research conducted by Fita Widiyatun, et al [7]. This research contains a physics calculator program made using Matlab GUI. The research method used in this research is literature study and development researchwith 4D approach (define, design, develop, disseminate). Testing in this research is in the form of matching or comparing the value of manual calculations with calculations generated by the GUI. The result of this research is a GUI that can be used to calculate momentum, impulse, and collision.

Research conducted by Hery Satria, et al [2]. This research contains Simulation of Chebyshev Type 1 Analog Filter Design using Matlab. This study aims to be able to produce a Chebyshev filter response and obtain the size of the inductor and capacitor needed to design the filter. The simulation results obtained Chebyshev's response for LPF, HPF, BPF and BSF and in accordance with the Chebyshev filter response theory.

RESEARCH METHOD

This research is a research that develops a program to build a GUI that can be used to filter signals with IIR filters. This research method consists of several main steps that will be described in detail in the following subsections:

Use of Matlab

Matlab is a software capable of performing mathematical computations, analyzing data, developing algorithms, performing simulations and modeling that produce graphic displays and graphical interfaces. The data types detected in Matlab programming are only two, namely numeric and string [8]. In the engineering college environment, Matlab is a standard tool for introducing and developing the presentation of mathematical, engineering and scientific materials. In industry, Matlab is the tool of choice for high-productivity research, development and analysis [7].

Matlab Graphical User Interface (GUI) Development

Matlab has a feature that allows users to create and design an attractive interface, this feature is GUIDE (Graphical User Interface Development Environment) [9]. In its use to be able to create a GUI, users need to type the GUIDE command in the Command Window in Matlab. The following are some components used in the GUI along with their functions [10]:

1. Push Button: execution button, if clicked will execute a command and display the results.
2. Edit Text: to enter input and display text results.
3. Axes: displays graphs or images.
4. Static Text: creates label text.
5. List box and Pop up Menu: opens a list of choices, and selects a choice.
6. Panel: used to group related GUI elements

Stand-Alone Feature

Stand-alone is a Matlab feature that allows the GUI files created to be run independently without requiring Matlab to be installed on the user's computer[11]. Stand-alone requires the Matlab Runtime (MCR) to be installed to function, but does not require a full Matlab license. Using the Matlab Compiler, users can package Matlab functions into executables (.exe) that can be distributed to others, allowing distribution of applications without revealing their source code.

Filter IIR Theory

IIR (Infinite Impulse Response) filter is a type of digital filter that has an infinite impulse response characteristic [12]. This means that the output of the filter can continue even though the input signal has stopped. IIR filters use feedback, where the current output is influenced by the previous input and the previous output, thus allowing the achievement of a steep frequency response with a smaller number of coefficients compared to FIR (Finite Impulse Response) filters. Some of these impulse responses can be modeled with rational system functions or with difference equations [13]. IIR filters have several advantages [5], namely IIR filters can produce steep frequency responses more efficiently than FIR filters. This allows IIR filters to focus more on the desired frequency band and reduce emphasis on frequency bands outside that range. In addition, IIR filters can be implemented to work in real-time in Digital Signal Processing (DSP) applications, where fast computing time is very important. These advantages make IIR filters widely used in various audio, telecommunications, and other signal processing applications.

IIR filters have several types, including Butterworth, Chebyshev I, Chebyshev II, Elliptic. Before creating a GUI, the creator must first know the equations used for several types of IIR filters. So, these equations will be entered into the program code in creating the GUI. The following are the equations for each type of filter:

a. Butterworth Filter

The Butterworth filter was introduced by a British scientist named Stephen Butterworth in 1930[14]. The Butterworth filter has a flat passband so that this filter provides the same amplification to almost all sinusoidal components in the passband area. The following is the Butterworth transfer function equation:

$$H(s) = \frac{1}{(s^2+sw_c+ w_c^2)^2} \quad (1)$$

b. Chebyshev I Filter

The Chebyshev I filter is derived from the polynomial equation introduced by Pafnuty Chebyshev. This Chebyshev I filter has ripples in the passband region so that this filter provides unequal amplification to the sinusoidal components in the passband region. The advantage of this filter is that it has a steeper slope in the transition band region so that it can have a narrow transition band at small orders [15]. The following is the Chebyshev I transfer function equation:

$$H(s) = \frac{H_0}{\sqrt{1+\epsilon^2 T_n^2(\frac{s}{w_c})}} \quad (2)$$

c. Chebyshev II Filter

The Chebyshev II filter is derived from the inverse Chebyshev polynomial equation. The Chebyshev II filter aims to create a filter with the same steep transition band characteristics as the Chebyshev Type I, but has a flat passband. As a consequence, this filter has ripples in the stopband region so that the attenuation in the stopband region is not good enough to filter out sinusoidal components that are not desired to be passed. The following is the Chebyshev II transfer function equation:

$$H(s) = \frac{H_0}{\sqrt{1+\frac{1}{\epsilon^2 T_n^2(\frac{w_c}{s})}}} \quad (3)$$

d. Elliptic Filter

Filter Elliptic juga dikenal dengan nama filter *Cauer* merupakan filter yang dengan *transition band* yang curam tetapi memiliki *ripple* pada *passband* dan *stopband*[16]. Besarnya *ripple* ini ditentukan pada saat filter ini dirancang. Jika *ripple* pada *passband* dibuat kecil maka *ripple* pada *stopband* akan besar, demikian sebaliknya. Berikut merupakan persamaan fungsi transfer *Elliptic*:

$$H(s) = \frac{H_0}{\sqrt{1+\epsilon^2 R_n^2(\frac{s}{w_c})}} \quad (4)$$

Types of Filters

In audio signal processing, filters can be classified based on the frequency range that is filtered or passed [17]. In general, Several types of filters can be identified, namely:

a. Lowpass Filter

Lowpass filter (LPF) is a filter that only passes sinusoidal components of low frequency input signals (< fc). This filter will stop high frequency sinusoidal components (> fc). Here is the equation formula of LPF:

$$H(s) = \frac{1}{s^2+s\frac{w_c}{Q}+w_c^2} \quad (5)$$

b. Highpass Filter

Highpass filter (HPF) is a filter that only passes sinusoidal components of high frequency input signals (> fc). High Pass Filter or usually abbreviated as HPF is a filter or frequency filter that can pass high frequency signals and inhibit or block low frequency signals[18]. The following is the equation formula for HPF:

$$H(s) = \frac{s^2}{s^2+s\frac{w_c}{Q}+w_c^2} \quad (6)$$

c. Bandpass Filter

Bandpass filter (BPF) is a filter that only passes sinusoidal components of the input signal whose frequency lies in a certain range (f1 < f < f2). This filter will stop sinusoidal components that are outside the frequency range (f < f1 or f > f2). Here is the equation formula of BPF:

$$H(s) = \frac{s\frac{w_c}{Q}}{s^2 + s\frac{w_c}{Q} + w_c^2} \quad (7)$$

d. Bandpass Filter

Bandstop filter (BSF) is a filter that only does not pass sinusoidal components of the input signal whose frequency lies in a certain range (f1 < f < f2). This filter will pass sinusoidal components that are outside the frequency range (f < f1 or f > f2). Here is the equation formula of BSF:

$$H(s) = \frac{s^2 + w_c^2}{s^2 + s\frac{w_c}{Q} + w_c^2} \quad (8)$$

Filter Order

The order of a filter is determined by the number of coefficients in its transfer function equation. If an analog filter is expressed in the form of H(s), then the order of the filter is the highest power of s in the equation H(s)[19]. A digital filter expressed in the form of an impulse response has an order of N-1, where N is the length of h(n). The order of the filter determines the similarity of the frequency response of a filter to the ideal response. The higher the order of a filter, the narrower its transition band[20]. On the other hand, the higher the order of a filter, the more complex its transfer function equation, which will result in a complicated implementation process for the filter.

Impulse Invariant

Impulse invariant is one of the methods of IIR filter to convert analog signal into digital signal[20]. IIR filter is usually expressed in the form of transfer function, H(z), as in the equation below. The final result of the design process of an IIR filter with the impulse invariant method is the value of the filter coefficient b_N, a_N, and gain, k.

$$H(z) = k\,\frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}} \quad (9)$$

Fast Fourier Transform

Fast Fourier Transform (FFT) is an efficient algorithm for calculating the Discrete Fourier Transform (DFT) and its inverse DFT used to analyze signals in the frequency domain, namely mapping time signals (time-domain) into frequency representation (frequency-domain)[21]. FFT speeds up this process by reducing computational complexity. The FFT algorithm works by dividing the signal into smaller parts, calculating the DFT for each part, and then combining the results efficiently. FFT is widely used in signal processing, spectrum analysis, and various other engineering fields. With FFT, complex signal analysis becomes faster and more practical, especially for large signals involving thousands or millions of samples.

3.1. Design of GUI Display

The interface of IIR filtering using GUI in Matlab allows users to input the required data and receive the results in an easy-to-understand manner. Figure 1 depicts a flowchart detailing the steps to use the GUI, which illustrates how users input data and receive the results. To enhance understanding, we introduced this flowchart earlier in the GUI design section.

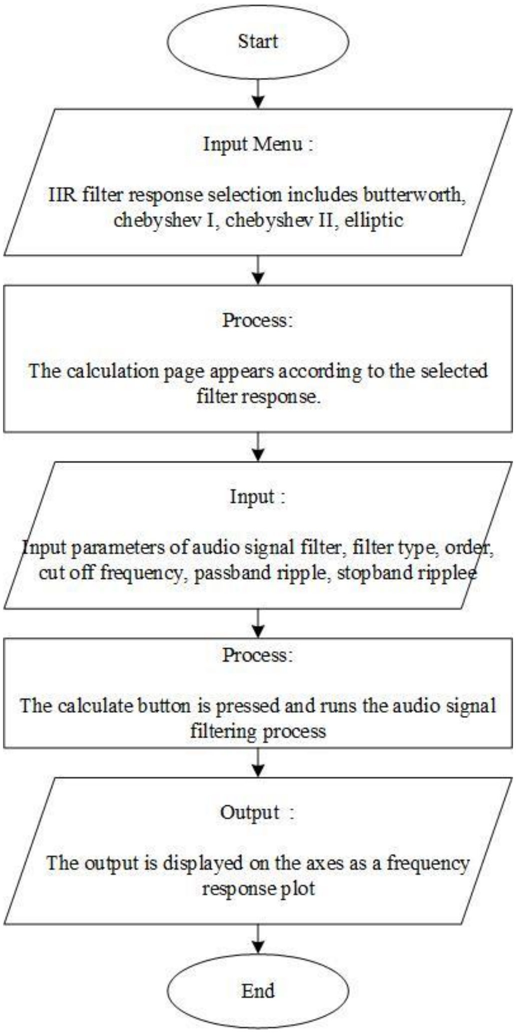

Figure 1. Flowchart of GUI usage

Create GUI Program

At this stage, the author develops the programming logic needed in the Matlab editor section to set the functions of each button on the GUI display. Through writing the right code, each button is given a specific task to do the job according to the needs, starting from inputting data, processing data, to saving data. This process ensures that the GUI can interact with users effectively and can perform the desired tasks to match the previously designed functionality.

Testing of the GUI

The purpose of this test is to verify whether the GUI works as planned and whether it is able to produce output values that are in accordance with expectations. The GUI testing process is carried out by comparing the audio signals tested using various different cutoff frequencies, while maintaining other parameters such as filter order and sampling frequency the same. Thus, the validity and accuracy of the software can be verified, so that the GUI can be trusted as a tool that can be used effectively in signal filtering. In addition, black box testing is also carried out on the function of each button, so that it can be seen whether the buttons can function optimally and there are no errors in the program. To test the appearance and functionality of the GUI, the researcher also conducted a test by giving a questionnaire to other students to determine the level of satisfaction with the GUI that had been produced.

RESULTS AND DISCUSSION

The resulting GUI is in the form of Menu and IIR GUI filter. The GUI for the IIR filter consists of three types of filters and can be seen in Figure 3-5. By providing input values in the input column, users can obtain a frequency response plot by clicking the “calculate” button. Data from the GUI can be saved to our computer in jpg and txt formats.

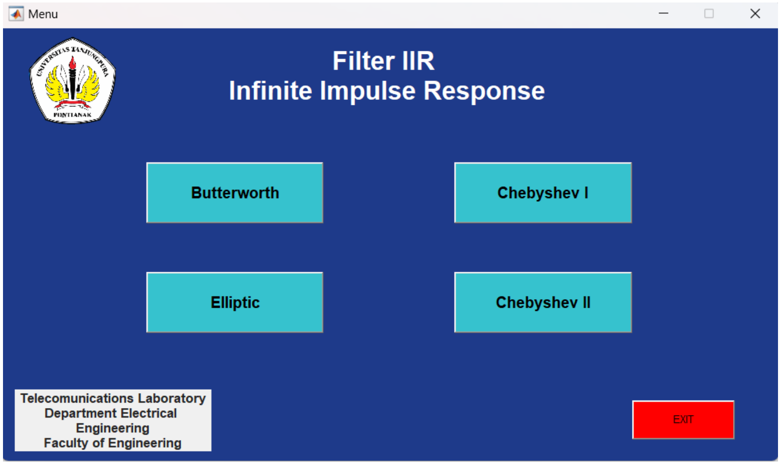

Figure 2. *GUI Menu Display*

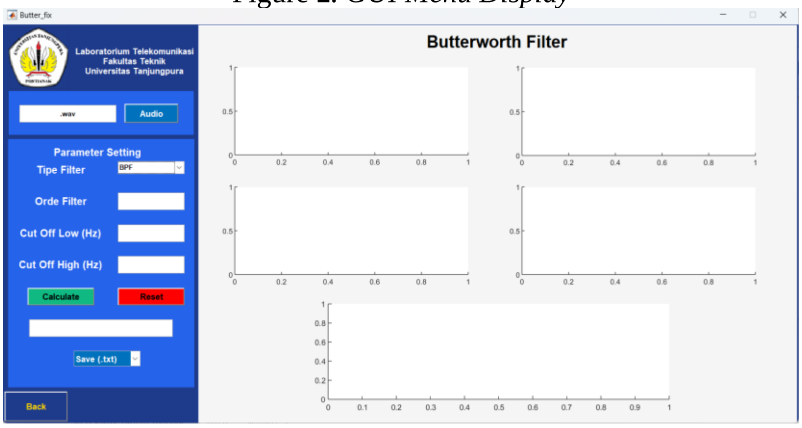

Figure 3. GUI Display of Butterworth Filter

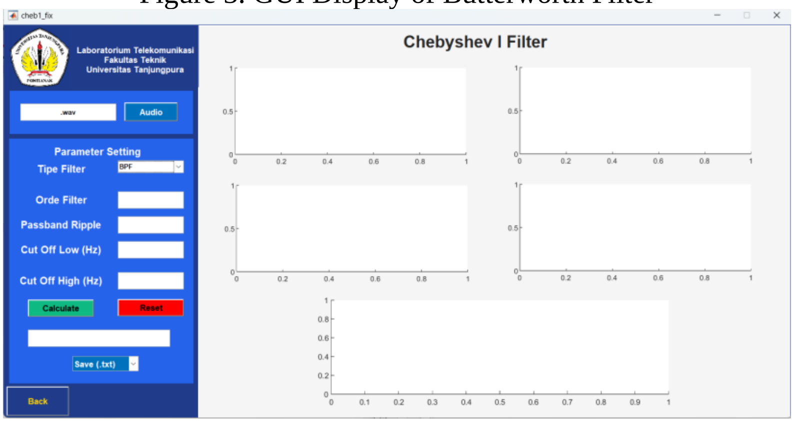

Figure 4. GUI Display of Chebyshev I Filter

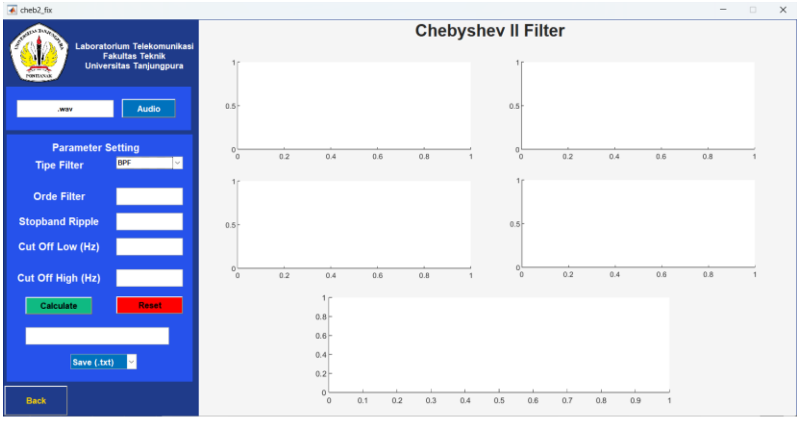

Figure 5. .I Display of Chebyshev II Filter

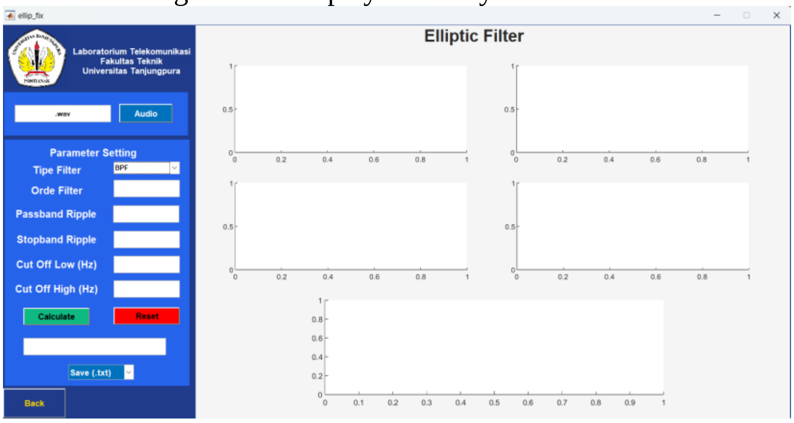

Figure 6. GUI Display of Elliptic Filter

Matlab GUI-based IIR Filter testing is done in several ways, namely filter validation testing, Blackbox testing, and GUI display testing using a questionnaire. In filter testing, it is done by comparing the audio signal being tested using various different cutoff frequencies while maintaining other parameters the same. It can be seen that the comparison of the output obtained from the GUI has the same characteristics as each filter. From the Blackbox test, it can be seen that all buttons and other components on the GUI can function according to their respective functions. Through the questionnaire that has been distributed, it can be seen that users are satisfied with the GUI produced.

Digital Filter Verification Test

Filter verification testing is done by comparing the audio signal being tested using various different cutoff frequencies while keeping other parameters the same. In this context, the filter characteristics obtained through the GUI indicate that the filter has been tested valid. The filter characteristics can be seen in Table 1 below:.

Table 1. IIR filter Characteristics

Filter Type	**Characteristics**
Butterworth	-Ripple in Passband. -Faster roll-off compared to Butterworth. -Flat stopband.
	LPF Ripples occur at low frequencies.
	HPF Ripple occurs at high frequencies.
	BSF Flat stopband, mid frequencies attenuated.

	BPF Ripple at allowed frequencies..
Chebyshev I	-Ripple in Passband. -Faster roll-off compared to Butterworth. -Flat stopband.
	LPF Ripples occur at low frequencies.
	HPF Ripples occur at high frequencies.
	BSF Flat stopband, mid frequencies are damped.
	BPF Ripple at allowed frequencies.
ChebyshevII	- Ripple in Stopband. -Flat passband (no Ripple). -Roll-off is faster than Butterworth but slower than Chebyshev I.
	LPF Flatter passband, sharper transitions.
	HPF Flatter passband, sharper transitions..
	BSF Ripple at damped frequencies.
	BPF Select a frequency between two cutoffs with a flat Passband.
Elliptic	-Ripple in Passband and Stopband. - Sharpest roll-off of all types. - High efficiency with low order design.
	LPF Ripple at low frequencies and fast transitions.
	HPF Ripple at high frequencies and fast transitions..
	BSF. Ripple on both sides of the frequency.
Filter Type	**Characteristics**
Elliptic	BPF Passband and Stopband have ripple.

Blackbox Test

Blackbox testing needs to be done to find out whether the buttons can function according to the provisions and ensure that there are no errors in the GUI program. This test is done by pressing each button on the GUI and seeing the output of the button press so that it can be known whether the button can function or not. The test results show that the GUI application is able to work according to its function, and the buttons can function optimally without any errors. The following results are shown in Table 2 below:

Table 2. Blackbox Test Result

Num	Action	Expected Result	Conclusion
1.	Click Audio button	Can retrieve audio files that are on computer storage.	Succeed
2.	Click Calculate button	Can perform audio signal filtering, the output produced is a frequency spectrum plot.	Succeed
3.	Click Save button	Select the GUI display storage format, namely jpg or txt format..	Succeed
4.	Click Listbox button	Selecting a different filter type	Succeed
5.	Click Reset button	Deletes the input output values so that the GUI display returns to its original appearance.	Succeed
6.	Click Back button	Displays the GUI menu and closes the currently opened GUI.	Succeed

Testing The GUI Display

In this testing phase, the GUI will be tested on prospective users. Testing was conducted on 30 other students who were prospective GUI users. Testing was conducted directly by giving questionnaires to students. This was done by the author to assess the level of user satisfaction with the GUI that had been produced. The questionnaire was compiled using a Likert scale on various aspects that can be seen in table 3 below:

Table 3. GUI satisfaction questionnare

Question	VS	S	N	D	VD
Content					
Question	**VS**	**S**	**N**	**D**	**VD**
- How satisfied are you with the GUI we have created?					
- To what extent our GUI aligns with your goals?					
- Do you feel that all the information you are looking for is available on our GUI?					
- How is the quality of image on our GUI?					
Design and Appearance Aspects					
Question	**VS**	**S**	**N**	**D**	**VD**
- How satisfied are you with the aesthetics and appearance of our GUI?					
- How easy it is to read and understand the GUI?					
- How logical and intuitive is the button layout on our GUI?					
Performance Speed Aspects					
Question	**VS**	**S**	**N**	**D**	**VD**
- How fast does our GUI load?					
- How often do you experience issues with the speed performance of our GUI?					
Functionality Aspects					
Question	**VS**	**S**	**N**	**D**	**VD**
- Do the buttons on the GUI work properly?					
- Are there any buttons that are not working in our GUI?					
- Are there any unused or unnecessary buttons in the					

context of this GUI?

	Question	VS	S	N	D	VD
-	How does the GUI notify an error if a button is pressed in a state where it shouldn't be? (for example, invalid input)					
-	What is your overall assessment of the functionality of the buttons on the GUI?					

Based on the results of the questionnaire filled out by 30 people, the calculation to calculate the average level of satisfaction of 30 people From the questionnaire, VS got a score of 5, S got a score of 4, N got a score of 3, D got a score of 2, and VD got a score of 1. So it can be calculated using the following equation:

$$AS= \frac{TNS}{TN} \tag{10}$$

In this equation, AS (Average Satisfaction of people) is calculated by dividing TNS (Total Questionnaire Score) by TN (Total Questionnaire). The results will provide an average value of the user's satisfaction level with the GUI. Furthermore, the level of satisfaction can be determined using the model defined by Kaplan and Norton. This model categorizes satisfaction levels into several stages, namely: (1) 1 – 1.79 = Verry Dissatisfied (VD); (2) 1.8 – 2.59 = Dissatisfied (D); (3) 2.6 – 3.39 = Neutral (N); (4) 3.4 – 4.19 = Satisfied (S); and (5) 4.2 – 5 = Very Satisfied (VS).

By using these criteria, we can measure the level of user satisfaction with the GUI has been created. The following results of the calculation of the level satisfaction of all aspects are shown in Table 4 below:

Table 4. People satisfaction level on all aspects

Content					
Question	**VS**	**S**	**N**	**D**	**VD**
Score	5	4	3	2	1
Q1	12	14	4	0	0
Q2	13	10	7	0	0
Q3	13	11	6	0	0
Q4	11	13	6	0	0
Total	49	48	23	0	0
SR	**4,21**				
Design and Appearance Aspects					
Question	**VS**	**S**	**N**	**D**	**VD**
Score	5	4	3	2	1
Q1	6	18	6	0	0
Q2	13	11	4	2	0
Q3	18	8	4	0	0
Total	37	37	14	2	0
SR	**4,21**				
Performance Speed Aspects					
Question	**VS**	**S**	**N**	**D**	**VD**
Score	5	4	3	2	1
Q1	19	8	3	0	0
Q2	9	12	9	0	0
Total	28	20	12	0	0

SR			**4,26**		
		Functionality Aspects			
Question	**VS**	**S**	**N**	**D**	**VD**
Score	5	4	3	2	1
Q1	15	11	4	0	0
Q2	15	11	4	0	0
Q3	16	10	4	0	0
Q3	12	12	6	0	0
Q4	16	12	2	0	0
Total	74	56	20	0	0
SR			**4,36**		

Table 5. Summary of visitor satisfaction levels

Aspect	**Average Satisfaction of visitors**	**Information**
Content	4,21	Very Satisfied
Design and Appearance	4,21	Very Satisfied
Performance Speed	4,26	Very Satisfied
Functionality	4,36	Very Satisfied
Total	4,26	Very Satisfied

Based on the results of the satisfaction survey on the GUI in Table 5, it can be concluded that most users are very satisfied with all aspects of this questionnaire. The GUI content is considered very relevant, the design and appearance of the GUI also reach a good level, the GUI can be used easily. The performance speed is quite high because it can process the input values given quickly. The functionality of the buttons is the highest because the buttons make it easy for users to obtain the desired information. However, there are several aspects that need more attention such as the GUI content which still needs improvement in level to achieve a better level of satisfaction. However, overall the survey results show that this GUI provides benefits for users to use to filter signals.

CONCLUSION

Based on the results of the experiment and testing of this study, it can be concluded that the design and simulation of the GUI of digital filters IIR Butterworth, Chebyshev I, Chebyshev II, and Elliptic with the types of Lowpass, Highpass, Bandstop, and Bandpass filters has been successfully carried out. Based on the GUI testing that has been done, the output of the GUI is the same as the characteristics of each type of filter from Lowpass, Highpass, Bandstop, and Bandpass so that the filtering results can be proven valid. Based on the Blackbox test, it can be seen that the buttons on the GUI function according to their respective functions so that there are no errors in the program and it is ready to use. Based on the survey that has been conducted, a score of 4.21 was obtained on GUI Content, 4.21 on GUI Design and Display, 4.26 on GUI Performance Speed, and 4.36 on GUI Functionality and according to the model set by Kaplan and Norton each of these scores can be categorized as Very Satisfactory. So it can be concluded that users are satisfied with the GUI that has been produced.

Acknowledgements

The author would like to thank the Chairperson and academic community of the Faculty of Engineering, Tanjungpura University and would like to thank all parties involved in the preparation of this journal. Hopefully the results of this journal can be useful for many people.

REFERENCES

M. Sri Rahayu Islamiyah and R. Saptono, “Implementasi Metode Transformasi Biliniear Pada Filter Digital Infinite Impulse Response (IIR) Menggunakan Raspberry Pi,” Jurnal Jaringan Telekomunikasi, vol. 11, no. 1, pp. 37–43, 2021.

Utama Herry Satria, “Simulasi Perancangan Filter Analog Chebyshev Tipe I Menggunakan MATLAB,” JET Jurnal Elektro Teknik, vol. 1, no. 1, pp. 39–45, 2021.

B. Maulydia, Mn. Zakaria, and waluyo, “Implementasi Metode Windowing Rectangular, Hamming, Hann, dan Kaiser pada Filter Finite Impulse Response (FIR) menggunakan Raspberry Pi,” Jurnal Jaringan Telekomunikasi (Jurnal Jartel), , vol. 10, pp. 96–101, 2020.

N. Manjula, N. P. Singh, and P. A. Babu, “An Efficient Designing of IIR Filter for ECG Signal Classification Using MATLAB,” Engineering Proceedings, vol. 34, no. 1, 2023, doi: 10.3390/HMAM2-14154.

U. Murdika, Y. Mulyani, J. Teknik Elektro Universitas Lampung, B. Lampung Jl Sumantri Brojonegoro No, and B. Lampung, “Perancangan Dan Simulasi Filter IIR Menggunakan Graphical User Interface (GUI),” ELECTRICIAN – Jurnal Rekayasa dan Teknologi Elektro, vol. 11, no. 3, pp. 1–5, 2017.

A. G. Utama, A. S. Nabila, A. K. Azzahra, D. Fadlun, and A. Assaidah, “Pembuatan Graphical User Interface (GUI) Matlab Untuk Demonstrasi Metode Windowing Pada Low Pass Filter (LPF) Finite Impulse Response (FIR),” Jurnal Penelitian Sains, vol. 25, no. 1, p. 34, Mar. 2023, doi: 10.56064/jps.v25i1.749.

Fita Widiyatun, “GUI Matlab Untuk Pembuatan Kalkultator Fisika Gerak Media Belajar Mahasiswa Teknik Informatika,” Prosiding Seminar Nasional Sains (SINASIS), vol. 2, no. 1, pp. 1–8, 2021.

D. Darmadi, Fitri Imansyah, and R. R. Yacoub, “Simulasi Eliminasi Noise Dengan Metode Transformasi Wavelet Berbantuan Graphical User Interface (GUI) Matlab,” Journal of Electrical Engineering, Energy, and Information Technology (J3EIT), vol. 8, no. 1, pp. 1–8, 2020.

D. T. V. Brian D. Hahn, Essential MATLAB for Engineers and Scientists, vol. 7. 2019.

T. Annisa, R. R. Yacoub, and E. Kusumawardhani, “Design of Microstrip Antenna Dimensions Based on The Matlab Graphical User Interface (GUI),” Journal of Electrical Engineering, Energy, and Information Technology (J3EIT, vol. 12, no. 2, pp. 441–452, 2024, doi: 10.26418/j3eit.v12i2.79742

H. Ratu et al., Solusi Numerik Konstruksi Scribs & GUI Berbasis Matlab, vol. 1.

R. Anugrahwaty and I. Yadi, “Design and Simulation of Analog Modulation with the Infinite Impulse Response Method,” Journal Of Informatics And Telecommunication Engineering, vol. 4, no. 2, pp. 310–316, Jan. 2021, doi: 10.31289/jite.v4i2.4411.

D. Datta and H. S. Dutta, “High performance IIR filter implementation on FPGA,” Journal of Electrical Systems and Information Technology, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s43067-020-00025-4.

A. Bustamin and A. Ais Prayogi, “Perbandingan Kinerja Filter Butterworth Berdasarkan Spesifikasi Frekuensi untuk Pengolahan Sinyal Suara,” Techno.COM, vol. 18, no. 4, pp. 332–339, 2019.

L. Lidyawati, A. R. Darlis, and A. F. Tamba, “Implementasi Filter Infinite Impulse Response (IIR) Dengan Respon Butterworth Dan Chebyshev Menggunakan DSK TMS320C6713,” Jurnal Elektro dan Telekomunikasi Terapan, vol. 2, no. 1, Jul. 2016, doi: 10.25124/jett.v2i1.97.

Li Tan, Digital Signal Processing, vol. 1. Georgia,

J. Obichere, E. Mfonobong, A. Isdore Onyema, O. Jude-kennedy Chibuzo, M. Eleazar Benson, and P. Chiedozie Francis, “Deployment of A Programmed Graphic User Interface (GUI) For Analysing Digital Filters Using Matlab-Algorithm,” IRE Journals, vol. 6, no. 10, pp. 1–16, 2023, [Online]. Available: https://www.researchgate.net/publication/

F. Trisnawati, D. Ramdan, and T. Elektro, “High Pass Filter,” Repoteknologi.id, vol. 1, no. 3, pp. 1–8, 2021.

N. Husna Shabrina, M. Regina Ao, N. Laurencya, V. Lee, and T. Fisika, “Analisis Aplikasi Filter

FIR dan Filter IIR dalam Pra-pemrosesan Sinyal Elektroensefalografi," 40 ULTIMA Computing, vol. 12, no. 1, pp. 1–9, 2020.

Nemuel Daniel Pah, Pemrosesan Sinyal Digital 2018, vol. 1. 2018.

M. Satria Adhi, A. Adisoka Bimantara, D. Priambodo, H. Muslim Azhar, and A. Junaidi, "Simulasi Penghilangan Noise Pada Sinyal Suara Menggunakan Metode Fast Fourier Transform (FFT)," Journal of Informatics, Information System, Software Engineering and Applications, vol. 1, no. 2, pp. 20–025, 2016, doi: 10.20895/INISTA.V1I2.